Clipper

เภา สิงห์บุบผา

สร้างฟังก์ชันทางด้านกราฟิกด้วย ภาษาซี ให้คลิปเปอร์เรียกใช้

นำเสนอเทคนิคการเชื่อมโยงภาษาซีและการสร้างฟังก์ชันจากซีให้คลิปเปอร์เรียกใช้

บทความต่อไปนี้ไม่ใช่เป็นการใช้ไลบรารีที่สร้างขึ้นมาให้คลิปเปอร์เรียกใช้เหมือนกับในบทความตอนก่อน ๆ ของผู้เขียน แต่จะใช้ความรู้ทางด้านการโปรแกรมภาษาซีมาใช้ในการสร้างฟังก์ชันจากภาษาซีเพื่อให้คลิปเปอร์เรียกใช้ เมื่อสร้างฟังก์ชันขึ้นมามาก ๆ แล้วก็สามารถรวบรวมฟังก์ชันต่าง ๆ เหล่านั้นเข้าด้วยกันเป็นไลบรารีเพื่อความสะดวกในการเรียกใช้ฟังก์ชันต่อไป มูลเหตุของบทความนี้ก็คือว่า ไลบรารีต่าง ๆ ที่นำมาใช้ศึกษา เช่น DGE, CLIPON, Nuntucket Tools II, Visual Interface For Clipper และอื่น ๆ อีกหลายอย่างนั้น ซึ่งจากการที่คลิปเปอร์สามารถเชื่อมโยงกับเครื่องมือช่วยเหลือเหล่านั้นได้อย่างง่ายดาย ทำให้มีผู้สร้างไลบรารีขึ้นเพื่อนำมาประกอบให้คลิปเปอร์มีความสามารถและมีความสะดวกในการใช้งานมากขึ้น แต่ต่อไปหากเครื่องมือเหล่านั้นมีราคาเพิ่มมากขึ้น การที่เราจะจัดหามาคงต้องเสียค่าใช้จ่ายมากขึ้น ดังนั้นผู้เขียนจึงเขียนบทความนี้ขึ้นเพื่อให้ผู้อ่านได้รู้จักการเชื่อมโยงภาษาซีเข้ากับคลิปเปอร์ เพื่อเป็นการเพิ่มความสามารถบางอย่างที่คลิปเปอร์ไม่มี อีกทั้งเกิดความภูมิใจที่เราสามารถสร้างไลบรารีขึ้นมาใช้ได้เอง ซึ่งผู้เขียนจะพยายามอธิบายให้ผู้อ่านสามารถนำไปประยุกต์ใช้ให้ได้ ส่วนไลบรารีที่ผู้เขียนกล่าวถึงนั้นคงจะนำมาลงบทความต่อ ๆ ไป

การติดตั้งซีคอมพายเลอร์เพื่อการใช้งาน

คอมพายเลอร์ภาษาซีที่ผู้เขียนแนะนำให้ใช้สำหรับบทความนี้คือ Microsoft C 5.00 และอีกตัวหนึ่งคือ Microsoft Quick C 2.5 ซึ่งตัวหลังนี้ผู้เขียนใช้อยู่เนื่องจากชอบ Help ที่มีประสิทธิภาพ อีกทั้งมีความเข้ากันได้ดีกับคอมพายเลอร์คลิปเปอร์ ในการติดตั้งนั้นให้ทำการติดตั้งไลบรารีแบบ Large Model ด้วย เพราะในการลิงค์นั้นจะต้องใช้ไลบรารี Llibce ของ Microsoft Quick C หรือ Llibca ของ Microsoft C และที่สำคัญคือต้องมีโปรแกรม LINK.EXE เพื่อใช้ในการลิงค์ และ LIB.EXE เพื่อใช้ในการจัดการไลบรารี

ซึ่งในขั้นตอนติดตั้งซี ก็จะได้โปรแกรม LINK.EXE และ LIB.EXE เพราะในการลิงค์ภาษาซีเข้ากับคลิปเปอร์นั้นจะลิงค์โดยใช้ LINK.EXE จะไม่ใช้ RTLINK ของคลิปเปอร์ ผู้เขียนก็ขอบอกกับผู้อ่านว่า ไลบรารีที่จำหน่ายเพื่อใช้ร่วมกับคลิปเปอร์นั้นโดยมากจะเขียนจากภาษาซีและแอสเซมบลีแล้วนำมาจำหน่าย ซึ่งเราก็สามารถเขียนขึ้นมาเองได้

การปรับสภาวะแวดล้อมให้คลิปเปอร์และซี

เมื่อติดตั้งคอมพายเลอร์ซีแล้ว ผู้อ่านต้องปรับสภาวะแวดล้อมเพื่อให้คอมพายเลอร์ทั้งสองทำงานได้หรือที่เรียกว่า Set Environment โดยการปรับพาธและกำหนดสัปไดเรคทอรีให้คอมพายเลอร์และโปรแกรม LINK สามารถค้นหาไฟล์ INCLUDE, OBJ และ LIB ตัวอย่างการปรับสภาวะแวดล้อมนี้ใช้สำหรับคลิปเปอร์ 5.01 หรือ 5.2 และ Microsoft Quick C 2.5 โดยจะอธิบายทั้งการปรับแต่ง CONFIG.SYS และ AUTOEXEC.BAT สำหรับซีพียูระดับ 386-SX และหน่วยความจำตั้งแต่ 1 เมกะไบต์ ขึ้นไปเพื่อให้เหลือหน่วยความจำ CONVENTION มากที่สุด

ผู้เขียนใช้ EMM386.EXE ในการบริหารหน่วยความจำ สำหรับบางท่านที่ใช้ SMARTDRV.EXE ในการทำดิสค์แคชให้ปรับ BUFFERS=10 และค่าของสแต็กอาจเปลี่ยนแปลงได้ เช่น STACKS=9,256

ผู้เขียนใช้ MS DOS เวอร์ชัน 6.00 การทำงานของไฟล์ AUTOEXEC.BAT ในรูปที่ 2 มีดังนี้

```
DEVICE=C:\DOS\HIMEM.SYS
DEVICE=C:\DOS\EMM386.EXE NOEMS
DOS=HIGH,UMB
BUFFERS=40
FILES=40
STACKS=0,0
```

รูปที่ 1 ไฟล์ CONFIG.SYS

```
01 : @ ECHO OFF
02 : PATH=C:\CLIPPER\BIN;C:\DOS;C:\QC25\BIN;C:\DOS\TSM
03 : Set LIB=C:\CLIPPER\LIB;C:\QC25\LIB
04 : Set INCLUDE=C:\CLIPPER\INCLUDE;C:\QC25\INCLUDE
05 : SET OBJ=C:\CLIPPER\OBJ
06 : SET TEMP=C:\DOS\TEMP
07 : LH NG
08 : LH MOUSE
09 : LH DOSKEY
10 : VGAMODE V
11 : MODE CON: RATE=32 DELAY=1
```

รูปที่ 2 ไฟล์ AUTOEXEC.BAT

1. ในบรรทัดที่ 02 เป็นการเซตพาธให้คลิปเปอร์และซี

2. บรรทัดที่ 03-05 เป็นการติดตั้งตัวแปรให้คอมพายเลอร์ได้เรียกหา Library, Include และ Object ในตอนคอมพายล์และลิงค์ทั้งภาษาซีและคลิปเปอร์

3. บรรทัดที่ 07 เป็นการโหลดนอร์ตันไกด์ซึ่งกินหน่วยความจำ 70 กิโลไบต์ออกจากหน่วยความจำคอนเวนชัน

4. บรรทัดที่ 08 และ 09 เป็นการโหลดโปรแกรมเม้าส์และดอสคีย์ขึ้นหน่วยความจำบนทั้งสอง โปรแกรมฝังตัวนี้ใช้หน่วยความจำประมาณ 14 กิโลไบต์ ซึ่งเมื่อทำเช่นนี้จะทำให้เหลือหน่วยความจำคอนเวนชันถึง 620 กิโลไบต์ ทั้งนี้ก็แล้วแต่แต่ละเครื่องอาจแตกต่างกันออกไป อีกทั้งโปรแกรม DOSKEY นั้นมีความจำเป็นเนื่องจากในการคอมพายล์และลิงค์นั้นจะเป็นคำสั่งที่มีพารามิเตอร์มาก ผู้เขียนจึงใช้ความสามารถในการเรียกใช้คำสั่งเดิมมาช่วยให้การทำงานเร็วขึ้น

5. บรรทัดที่ 11 เป็นการกำหนดความเร็วของคีย์บอร์ดให้ทำงานเร็วขึ้นและมีการหน่วงที่น้อยลง

6. บรรทัดที่ 10 เป็นไดรเวอร์ของการ์ดวีจีเอ เนื่องจากการ์ดวีจีเอจะทำงานได้ดีต้องมีไดรเวอร์ควบคู่ไปด้วย

สำหรับการ์ดแต่ละรุ่นจะมีไดรเวอร์ที่ต่างกันไป ศึกษาได้จากคู่มือการ์ดนั้น

ที่ผู้เขียนต้องอธิบายอย่างละเอียดนี้เพื่อช่วยให้ผู้อ่านประสบผลสำเร็จในการศึกษาโปรแกรม เพราะบางท่านทำการคอมพายล์และลิงค์โปรแกรมไม่สำเร็จก็เพราะสิ่งเล็กน้อยบางอย่างเท่านั้น เช่น หน่วยความจำไม่พอ ลิงค์แล้วหาออปเจ็กต์ไฟล์ไม่เจอ ไม่พบไลบรารีบ้าง ทำให้เกิดความท้อถอย ผู้เขียนมีความรู้เท่าไรก็อยากช่วยให้ผู้อ่านที่เขียนโปรแกรมโดยคลิปเปอร์ได้มีความรู้เพิ่มขึ้นให้มากที่สุด และสำหรับผู้อ่านที่ไม่เคยศึกษาภาษาซีก็ไม่ต้องท้อถอย เพราะผู้เขียนจะเขียนในแนวทางที่ให้ภาษาซีเรียกใช้อินเตอร์รัพต์ของดอสและไบออส ทำให้ผู้อ่านสามารถประยุกต์หรือดัดแปลงได้ง่ายขึ้น อีกอย่างแนวทางในการเขียนโปรแกรมโดยคลิปเปอร์นั้นใกล้เคียงกับภาษาซีมากขึ้น ดังนั้นผู้อ่านที่ศึกษาการเขียนโปรแกรมคลิปเปอร์จนชำนาญก็จะสามารถศึกษาการเขียนภาษาซีหรืออ่านโปรแกรมภาษาซีได้ไวขึ้น

Clipper Type	C Type	Extend Function
character	char*	_parc()
	int	_parclen()
date	char*	_pards()
logical	int	_parl()
numeric	int	_parni()
	long	_parnl()
	double	_parnd()

รูปที่ 3 แสดงการส่งผ่านค่าจากคลิปเปอร์ไปยังซี

```
typedef double XDOUBLE;
/* parameter values */
extern char *_parc(int, ...);
extern unsigned int _parcsiz(int, ...);
extern unsigned int _parclen(int, ...);
extern int _parni(int, ...);
extern long _parnl(int, ...);
extern XDOUBLE _parnd(int, ...);
extern int _parl(int, ...);
extern char *_pards(int, ...);
```

รูปที่ 4 แสดงการส่งผ่านค่าจากคลิปเปอร์ไปยังซี

การส่งผ่านข้อมูลจากคลิปเปอร์ไปยังซี

เนื่องจากการกำหนดตัวแปรของคลิปเปอร์และซีไม่เหมือนกัน เช่น ในคลิปเปอร์กำหนดตัวแปรเป็นชนิดตัวเลข (numeric) แต่ในซีอาจกำหนดเป็น int, long หรือ double ในการกำหนดตัวแปรเพื่อส่งผ่านค่าจากคลิปเปอร์ไปยังซี จึงต้องปรับเปลี่ยนชนิดของตัวแปร ซึ่งกำหนดอยู่ในไฟล์ EXTEND.H ซึ่งต้องนำมาใช้ร่วมในการเขียนฟังก์ชันของซีที่ต้องการส่งผ่านค่าจากคลิปเปอร์ไปยังซีดังรูปที่ 1 ซึ่งมีการเปลี่ยนค่าดังรูปที่ 3

สำหรับการกำหนดการส่งผ่านค่านั้น จะถูกกำหนดในไฟล์ EXTEND.H ซึ่งผู้อ่านต้องอินคลูดไฟล์ในฟังก์ชันของซีที่เขียนขึ้นมาให้คลิปเปอร์เรียกใช้ดังรูปที่ 4

การส่งค่ากลับจากซีมายังคลิปเปอร์

การส่งค่ากลับจากภาษาซีมายังคลิปเปอร์ก็เช่นเดียวกับการส่งผ่านค่าจากคลิปเปอร์ไปยังซีที่ต้องมีการเปลี่ยนชนิดของข้อมูลดังรูปที่ 5

ให้ผู้อ่านดูจากไฟล์ EXTEND.H ในรูปที่ 6

การคอมพายล์และลิงค์

ในการเขียนฟังก์ชันภาษาซีเพื่อให้คลิปเปอร์เรียกใช้นั้นผู้เขียนใช้คอมพายเลอร์ Microsoft Quick C 2.5 โดยจะใช้เอดิเตอร์ QEditer เขียนซอร์สโค้ด การคอมพายล์จะใช้ Quick C Compiler Version 2.5 ซึ่งมีชื่อไฟล์ QCL.EXE อยู่ในสัปไดเรคทอรี C:\QC25\

C Type	Clipper Type	Extend Function
char*	character	_retc()
		_retclen()
	date	_retds()
int	logical	_retl()
	numeric	_retni()
long	numeric	_retnl()
double	numeric	_retnd()

รูปที่ 5 การส่งผ่านค่าจากซีมายังคลิปเปอร์

```
/* return a value */
extern void _retc(char far *);
extern void _retclen(char far *, unsigned int);
extern void _retni(int);
extern void _retnl(long);
extern void _retnd(XDOUBLE);
extern void _retl(int);
extern void _retds(char far *);
```

รูปที่ 6 ไฟล์ EXTEND.H

BIN ซึ่งเราได้กำหนดให้อยู่ในพาธแล้ว เพื่อสะดวกในการเรียกใช้ ส่วนในการลิงค์ไฟล์ที่ได้จากการคอมพายล์จากคลิปเปอร์และซีเราจะไม่ใช้ RTLINK แต่จะใช้ LINK.EXE ที่มีมากับดอสหรือที่ติดตั้งมากับคอมพายเลอร์ซีก็ได้

ในการคอมพายล์ซีนั้น นอกจากจะต้องอินคลูดไฟล์ EXTEND.H แล้ว ยังอาจต้องอินคลูดไฟล์ .H อื่นอีกบ้างตามความจำเป็นสำหรับการเขียนภาษาซี หากใช้ซีเรียกอินเตอร์รัพต์นั้นจะต้องอินคลูดไฟล์ DOS.H ด้วยทุกครั้งดังนี้

```
#include "extend.h"
#include "dos.h"

CLIPPER funcname( )
{
    .
    .
    .
}
```

ตรงบรรทัด CLIPPER funcname( ) เป็นการกำหนดชื่อฟังก์ชันในซีให้คลิปเปอร์เรียกใช้ เมื่อเขียนฟังก์ชันของซีเสร็จต้องคอมพายล์โดยกำหนดค่าในการคอมพายล์ดังนี้

หากให้สะดวกควรเขียนแบตช์ไฟล์ไว้ในชื่อ C.BAT เพื่อความรวดเร็วในตอนคอมพายล์ดังนี้

```
QCL /c /AL /Zl /Olt /Gs funcname.c
C.BAT
QCL /c /Al /Zl /Olt /Gs %1.c
```

ความหมายของการกำหนดค่าให้ผู้อ่านดูได้จาก help หรือคู่มือ Microsoft Quick C ในการลิงค์ให้กำหนดการลิงค์ดังนี้

```
LINK/NOE/SE:512 obj,,,llibce+userlib
```

จะเห็นได้ว่าในการลิงค์เราไม่ได้บอกว่าจะต้องลิงค์กับไลบรารี CLIPPER.LIB และ EXTEND.LIB โปรแกรมลิงค์จะรู้เองว่าต้องการไลบรารีใดและจะไปค้นหาตามพาธที่กำหนดใน AUTOEXEC.BAT ในการลิงค์นี้ต้องการไลบรารีของซีแบบ Large Model ที่ชื่อว่า LLIBCE.LIB ซึ่งเรากำหนดให้ลิงค์รวมกับไลบรารีที่เราสร้างขึ้นจากภาษาซี (userlib) ผู้เขียนใช้ LINK.EXE Version 3.69 ผู้อ่านควรใช้อย่าให้ต่ำกว่านี้

เริ่มแบบฝึกหัดแรก

เพื่อให้ผู้อ่านได้ทดสอบการเขียนฟังก์ชันในภาษาซีเพื่อให้คลิปเปอร์เรียกใช้ได้เพื่อเป็นแรงจูงใจในการศึกษาเรื่องนี้ต่อ ผู้เขียนขอเริ่มตัวอย่างแรกเลยโดยจะกล่าวถึงการนำเอาภาษาซีมาเรียกใช้อินเตอร์รัพต์ของดอสหรือไบออสในภายหลัง ตัวอย่างแรกผู้เขียนจะแสดงโปรแกรมที่กำหนดขนาดของเคอร์เซอร์ในรูปที่ 7 จะเป็นฟังก์ชันที่เขียนขึ้นในภาษาซี

```
#include "extend.h"
#include "dos.h"
CLIPPER setcur()
{
    union REGS regs;
    regs.h.ah=0x01;
    regs.h.ch=_parni(1);
    regs.h.cl=_parni(2);
    int86(0x10,&regs,&regs);
}
```

รูปที่ 7 ฟังก์ชันที่เขียนขึ้นในภาษาซีให้คลิปเปอร์เรียกใช้เพื่อการกำหนดขนาดของเคอร์เซอร์

```
// PROGRAM NAME : CTEST.PRG
// COMPILE      : CLIPPER ctest
// LINK         : LINK/NOE/SE:512 ctest,,,llibce+ctest
// RUN          : CTEST

          SETCUR(4,7)
          WAIT
          SETCUR(7,7)
          WAIT
          SETCUR(1,7)
          WAIT
          RETURN
```

รูปที่ 8 โปรแกรม CTEST.PRG

เมื่อพิมพ์ซอร์สโค้ดแล้วให้เซฟข้อมูลโดยตั้งชื่อแฟ้มว่า CURSET.C แล้วคอมพายล์ด้วยคำสั่งนี้

```
QCL /c /AL /Zl/Olt/Gs curset.c
```

หรือใช้แบตช์ไฟล์คอมพายล์ดังนี้

```
C curset
```

เมื่อคอมพายล์แล้วจะได้ไฟล์ curset.obj เพื่อนำไปลิงค์ใช้งานต่อไป เพื่อความสะดวกเราจะรวบรวมฟังก์ชันที่เขียนจากภาษาซีไปเป็นไลบรารีเพื่อสะดวกในการเรียกใช้ เมื่อเราคอมพายล์ curset.c แล้วจะได้ curset.obj เราจะเปลี่ยนไปเก็บเป็นไลบรารีโดยคำสั่งนี้

```
LIB ctest +curset
```

จะได้ไฟล์ CTEST.LIB ที่คลิปเปอร์สามารถเรียกใช้ได้ สำหรับโปรแกรมที่เขียนขึ้นสำหรับคลิปเปอร์แสดงในรูปที่ 8 ซึ่งเขียนขึ้นมาเพื่อเป็นตัวอย่างการเรียกใช้ฟังก์ชัน SETCUR() จากภาษาซีที่เขียนขึ้นจากรูปที่ 7 ให้คอมพายล์โปรแกรมในรูปที่ 8 ด้วยคลิปเปอร์ เมื่อคอมพายล์แล้วให้ลิงค์ด้วยโปรแกรม LINK.EXE ดังคอมเมนต์ในรูปที่ 7 เมื่อทดลองรันโปรแกรม CTEST.EXE แล้วสังเกตลักษณะของเคอร์เซอร์

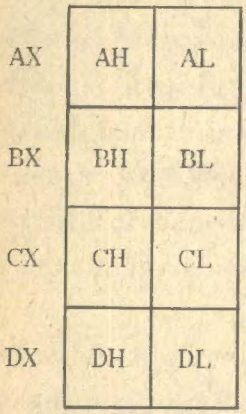
สำหรับบทความในตอนนี้ก็เพื่อให้ผู้อ่านมองเห็นขั้นตอนคร่าว ๆ ในการเชื่อมโยงซีกับคลิปเปอร์ว่าไม่ใช่เรื่องยากเย็นอะไรนัก (การใช้ LIB.EXE ให้ศึกษาการใช้จากคู่มือดอสหรือดูที่ไมโครฯ ฉบับที่ 99 หน้า 316-317)

การเรียกใช้ฟังก์ชัน DOS และ BIOS

ในตอนนี้ผู้เขียนขอตกลงกับผู้อ่านให้อดทนอ่านสักหน่อย เพราะหากก่อย ๆ พิจารณาจะเป็นเรื่องที่ไม่ยากเย็นนัก ผู้เขียนขอแนะนำผู้อ่านให้ศึกษาหนังสือที่เขียนเกี่ยวกับการเรียกใช้อินเตอร์รัพต์ของซี เช่น คู่มือโปรแกรมภาษาซีสำหรับเบื้องต้น (ซีเอ็ดยูเคชั่น) และควรมีคู่มือในการอ้างอิงเกี่ยวกับอินเตอร์รัพต์ เช่น แอดวานซ์เอ็มเอสดอส (ซีเอ็ดยูเคชั่น) สำหรับการอ้างอิงการเรียกใช้อินเตอร์รัพต์

ภาษาซีสามารถเชื่อมโยงกับระบบปฏิบัติการอย่างดอสและระบบไบออสของเครื่องโดยการเรียกใช้รีจิสเตอร์ของ 8088 หรือ 8086 สำหรับแผนภาพของรีจิสเตอร์ 8086 เป็นดังรูปที่ 9

รีจิสเตอร์จะมีทั้งขนาด 16 บิต และ 8 บิต สำหรับรีจิสเตอร์ 16 บิต ประกอบด้วยรีจิสเตอร์ AX,BX,CX และ DX ส่วนรีจิสเตอร์ 8 บิตประกอบด้วย AH, AL, BH, BL, CH, CL, DH



รูปที่ 9 แสดงรีจิสเตอร์ 8086

และ DL การกำหนดค่ารีจิสเตอร์ต่าง ๆ นี้สามารถทำได้ในภาษาซีตามแบบอย่างภาษาแอสเซมบลี โดยการกำหนดค่ารีจิสเตอร์เป็นคำสั่งที่ขึ้นกับฮาร์ดแวร์ของระบบคอมพิวเตอร์นั้น และการกำหนดค่ารีจิสเตอร์ต้องดูค่าอินเตอร์รัพต์ของดอสและไบออสจากหนังสืออ้างอิง ภาษาซีจะใช้การกำหนดโครงสร้างของข้อมูลแทนค่ารีจิสเตอร์ โครงสร้างที่กำหนดค่าของรีจิสเตอร์อาจเป็นขนาด 16 บิตหรือ 8 บิตโดยใช้คำสั่งยูเนียน ดังนี้

```
union REGS
        {
        struct WORDREGS x;
        struct BYTEREGS h;
        }
```

ดังนั้นการกำหนดชนิดของตัวแปรให้เป็นยูเนียนจึงต้องกำหนดดังนี้ (สังเกตจากรูปที่ 7)

```
union REGS regs;
```

เป็นการกำหนดตัวแปร regs เป็นชนิดตัวแปรยูเนียน REGS ซึ่งมีโครงสร้างชนิด WORDREGS และ BYTEREGS อยู่ในหน่วยความจำเดียวกัน โดยมีสมาชิกของโครงสร้างถูกกำหนดเป็นตัวแปรเลขจำนวนเต็มเพื่อแทนรีจิสเตอร์ของคำและไบต์ดังนี้

```
struct WORDREGS
        {
        unsigned int ax;
        unsigned int bx;
        unsigned int cx;
        unsigned int dx;
        unsigned int si;
        unsigned int di;
        unsigned int cflag;
        }
struct BYTEREGS
        {
        unsigned int ah,al;
        unsigned int bh,bl;
        unsigned int ch,cl;
        unsigned int dh,dl;
        }
```

การกำหนดค่าของตัวแปร regs จึงเป็นตัวแทนได้ทั้งไบต์และคำ เช่น ในรูปที่ 7 กำหนดเป็นไบต์ เป็นต้น

การเรียกฟังก์ชันดอสและไบออส

โปรแกรมภาษาซีที่จะเขียนขึ้นมาเพื่อเรียกใช้บริการของดอสและไบออสต้องใช้แฟ้มข้อมูลหัว (เฮดเดอร์ไฟล์) ที่ชื่อ dos.h ของภาษาซีเพื่อการกำหนดตัวแปร REGS, WORDREGS และ BYTEREGS รวมทั้งการเชื่อมโยงกับ int86( ) และเฮดเดอร์ไฟล์ที่ชื่อ extend.h ที่ให้มากับคลิปเปอร์ (อยู่ใน C:\CLIPPER\INCLUDE) เพื่อกำหนดการส่งและรับค่าตัวแปรระหว่างคลิปเปอร์และซี สำหรับรูปแบบของฟังก์ชัน int86( ) เป็นดังนี้

```
int86 (INT, &regs, &regs);
```

โดย INT เป็นค่าอินเตอร์รัพต์ที่ใช้เป็นเลขฐานสิบหก ส่วน &regs ทั้งสองตัวเป็นพอยน์เตอร์ของตัวแปรชนิด REGS สำหรับการกำหนดค่าให้ศึกษาจากรูปที่ 7

การกำหนดชนิดของจอภาพเพื่อเข้าสู่การเขียนกราฟิก

ในคอมพายเลอร์คลิปเปอร์ไม่มีฟังก์ชันที่เปลี่ยนการแสดงผลจากเท็กซ์โหมดสู่กราฟิกโหมด ในฉบับที่แล้ว ๆ มาผู้เขียนใช้ไลบรารี DGE ในการเปลี่ยนโหมดการแสดงผล เช่น ฟังก์ชัน SETHIRES( ) แต่ในตอนนี้เราจะใช้ภาษาซีในการเปลี่ยนโหมดแทนแล้วให้คลิปเปอร์เรียกใช้ ก่อนที่เราจะเข้าสู่การเขียนโปรแกรม เราจะมาศึกษาการจัดการส่วนแสดงผลคือ ROM BIOS video driver ที่เราสามารถเรียกใช้ผ่านอินเตอร์รัพต์ 10H (หากต้องการรายละเอียดมากขึ้นให้ศึกษาจากหนังสือแอดวานซ์เอ็มเอสดอสของซีเอ็ดฯ) ดังในรูปที่ 10

ผู้เขียนยกมาแค่ 2 โหมดเท่านั้น คือ โหมด 3 และโหมด 18 เพื่อการเข้าสู่กราฟิกโหมดและกลับสู่เท็กซ์โหมด หากผู้อ่านต้องการเข้าสู่การแสดงผลในโหมดอื่นให้ดูหมายเลขโหมดการแสดงผลและความสามารถของส่วนแสดงผลของคอมพิวเตอร์แต่ละเครื่องด้วย สำหรับการเข้าสู่กราฟิกโหมด 12H นั้นในเลขฐานสิบคือโหมดการแสดงผลที่ 18 นั่นเอง

รูปที่ 11 เป็นการเขียนฟังก์ชันโดยภาษาซี ซึ่งจะรับค่าพารามิเตอร์จากการเรียกใช้ของคลิปเปอร์โดยมีรูปแบบการเรียกใช้ในคลิปเปอร์ดังนี้

```
SetMode(nMode)
```

nMode คือโหมดการแสดงผลของจอภาพที่สามารถแสดงได้ ให้ใส่

```
INT 10H(16)
Function 00H0
Set video mode
จุดประสงค์ : เลือกโหมดการแสดงผลทางจอภาพ
เรียกการทำงานโดย :
   AH   =   00H
   AL   =   โหมดการแสดงผล
            03H สำหรับความละเอียด 80x25  สีของจอภาพ Color text 16 สี
            12H สำหรับความละเอียด 640x480 สีของจอภาพ Color graphic 16 สี
```

รูปที่ 10

```
01:  /***************************************
02:  * Function Name.... SetMode()
03:  * Programmer....... Bhao Singboobpha
04:  * Date............. 16-04-37
05:  * Syntax........... SetMode(nMode)
06:  * Return........... no return
07:  * Description...... set video mode
08:  ***************************************/
09:  #include "extend.h"
10:  #include "dos.h"
11:
12:  CLIPPER SETMODE()
13:  {
14:     union REGS regs;
15:     regs.h.ah=0x00;
16:     regs.h.al=_parni(1);
17:     int86( 0x10, &regs, &regs);
18:  }
```

รูปที่ 11 ฟังก์ชัน SetMode( ) ชื่อไฟล์ setmode.c

เป็นเลขฐานสิบ การทำงานของฟังก์ชันในรูปที่ 11 มีดังนี้

บรรทัดที่ 9-10 เป็นการอินคลูดเฮดเดอร์ไฟล์

บรรทัดที่ 12 เป็นการตั้งชื่อฟังก์ชัน

บรรทัดที่ 14 เป็นการกำหนดชนิดของตัวแปร

บรรทัดที่ 15 กำหนดให้รีจิสเตอร์ AH มีค่าเท่ากับ 00

บรรทัดที่ 16 กำหนดให้รีจิสเตอร์ AL รับค่าพารามิเตอร์จากการเรียกใช้ของคลิปเปอร์เพื่อการกำหนดโหมดการแสดงผล โดยรับค่ามาจาก _parni(1)

บรรทัดที่ 17 เรียกใช้ฟังก์ชัน int86( )

เมื่อผู้อ่านพิมพ์ซอร์สโค้ดจนเสร็จแล้วให้เซฟข้อมูลโดยตั้งชื่อไฟล์ว่า setmode.c ทั้งนี้ชื่อไฟล์ไม่จำเป็นต้องเหมือนกับชื่อฟังก์ชัน แต่เพื่อไม่ให้สับสนจึงควรกำหนดชื่อให้ใกล้เคียงหรือเหมือนกัน ให้คอมพายล์ดังนี้

```
QCL /c /AL /Zl /Olt /Gs setmode.c
```

หรืออาจใช้แบตช์ไฟล์ C.BAT คอมพายล์ก็ได้ดังนี้

```
c setmode
```

เมื่อคอมพายล์แล้วจะได้ไฟล์ setmode.obj ให้รวมเป็นไลบรารีที่ชื่อ GRAPHIC.LIB โดยคำสั่งนี้

```
LIB GRAPHIC +SETMODE
```

จะได้ไฟล์ GRAPHIC.LIB ที่ประกอบด้วยฟังก์ชัน SetMode( ) เพียงฟังก์ชันเดียว ซึ่งเมื่อเราได้เขียนฟังก์ชันจากภาษาซีแล้ว เราสามารถนำมาเก็บรวบรวมไว้ในไลบรารี GRAPHIC.LIB เพิ่มมากขึ้นต่อไป

เขียนโปรแกรมคลิปเปอร์เพื่อเรียกใช้ฟังก์ชันภาษาซี

เราจะมาเขียนโปรแกรมง่าย ๆ เพื่อส่งค่าพารามิเตอร์ไปให้ฟังก์ชันภาษาซีดังรูปที่ 4 ซึ่งเป็นการเข้าสู่กราฟิกโหมดที่ 18

เมื่อคอมพายล์ในคลิปเปอร์โดยคำสั่ง

```
CLIPPER Tograph
```

แล้วลิงค์โดยคำสั่ง

```
LINK/NOE/SE:512 tograph,,,llibce+graphic
```

จะได้ไฟล์ TOGRAPH.EXE เมื่อรันโปรแกรมแล้วจะเข้าสู่กราฟิกโหมด (MODE 18) โดยสังเกตว่าจะไม่มีเคอร์เซอร์กะพริบที่เครื่องหมายเตรียมพร้อม เนื่องจากอยู่ในสภาวะการแสดงผลแบบกราฟิก เราควรเตรียมโปรแกรมเพื่อกลับสู่เท็กซ์โหมดไว้ก่อนด้วย ดังแสดงในรูปที่ 13

เมื่อคอมพายล์ในคลิปเปอร์โดยคำสั่ง

```
CLIPPER Totext
```

แล้วลิงค์โดยคำสั่ง

```
LINK/NOE/SE:512 totext,,,llibce+graphic
```

จะได้ไฟล์ TOTEXT.EXE เมื่อรันจะกลับสู่เท็กซ์โหมด (MODE 3)

เปรียบเทียบกับโปรแกรมในรูปที่ 7 โปรแกรมในรูปที่ 7 จะเรียกอินเตอร์รัพต์หมายเลข 10H เหมือนกัน แต่เป็นฟังก์ชันที่ 01H ซึ่งมีรายละเอียดดังรูปที่ 14

ดังนั้นในการกำหนดขนาดของเคอร์เซอร์จึงต้องกำหนดค่าตำแหน่งบรรทัดที่เริ่มเคอร์เซอร์ให้รีจิสเตอร์ CH และกำหนดค่าตัวที่สองตำแหน่งบรรทัดที่สิ้นสุดเคอร์เซอร์ให้รีจิสเตอร์ CL โดยโปรแกรมในรูปที่ 7 จะรับค่าพารามิเตอร์จาก _parni(1) และ _parni(2) ที่ส่งค่าจากการเรียกใช้ฟังก์ชัน SETCUR( ) ในโปรแกรมรูปที่ 8 ที่จะต้องกำหนดการส่งค่า 2 ตัวเช่น SETCUR(4, 7)

บทส่งท้าย

ผู้เขียนหวังว่าบทความที่เขียนขึ้นนี้คงช่วยให้ท่านผู้อ่านมีความเข้าใจการเชื่อมโยงภาษาซีมากขึ้น ขอให้ผู้อ่านค่อย ๆ พยายามทำความเข้าใจ ในตอนต่อไปผู้เขียนจะได้แนะนำการเขียนฟังก์ชันภาษาซีเกี่ยวกับการลงจุด เส้น กรอบสี่เหลี่ยม วงกลม ในโหมดกราฟิก การอินเตอร์เฟสเม้าส์เข้ากับคลิปเปอร์ในโหมดเท็กซ์ และการติดต่อระหว่างเมนูของโปรแกรมที่เขียนด้วยคลิปเปอร์กับเม้าส์ ซึ่งจะทำให้ผู้อ่านได้เกิดความเข้าใจมากขึ้น *สวัสดีครับ*

```
// Program     : TOGRAPH.PRG
// Description : set mode from text TO graphic 18
SetMode(18)
RETURN
```

รูปที่ 12 โปรแกรม TOGRAPH.PRG

```
// Program     : TOTEXT.PRG
// Description : set mode from graphic TO text
SetMode(3)
RETURN
```

รูปที่ 13 โปรแกรม TOTEXT.PRG

```
INT 10H(16)
Function 01H(1)
Set cursor type
จุดประสงค์ กำหนดขนาดเคอร์เซอร์
เรียกทำงานโดย:  AH  = 01H
                CH  = ตำแหน่งบรรทัดที่เริ่มเคอร์เซอร์
                CL  = ตำแหน่งบรรทัดที่สิ้นสุดเคอร์เซอร์
```

รูปที่ 14